

## AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application.

Claims 1-19 (Cancelled)

20. (Currently Amended) A method comprising:

receiving [[a packed data]] an instruction that specifies memory locations of a first full-width packed data operand having a plurality of data elements and a second full-width packed data operand having a corresponding plurality of data elements;

[[substantially simultaneously]] accessing the first full-width packed data operand and the second full-width packed data operand from the memory locations;

dividing the first full-width packed data operand into a first subset of data elements and a second subset of data elements;[[and]]

dividing the second full-width packed data operand into a third subset of data elements and a fourth subset of data elements;

performing an operation specified by the [[packed data]] instruction on the first and third subsets of data elements to generate a first resulting one or more data elements;

delaying the second and fourth subsets of data elements;

after said delaying, performing an operation specified by the [[packed data]] instruction on the second and the fourth subsets of data elements to generate a

second resulting one or more data elements, wherein performing the operation specified by the instruction on the second and the fourth subsets comprises setting at least one data element to a predetermined value; and

storing the first and the second resulting data elements in a common packed data operand.

Claim 21 (Cancelled)

22. (Currently Amended) The method of claim 20, wherein dividing the first full width packed data operand includes dividing a 128-bit packed data operand into a first 64-bit segment of two low order data elements and a second 64-bit segment of two high order data elements.

23. (Currently Amended) A processor comprising:

[[a packed data instruction to specify an operation on a plurality of data elements of at least one packed data operand;]]

a decoder to receive a partial-width packed data instruction specifying an operation on a plurality of data elements of at least one packed data operand, the decoder to generate a first micro instruction and a second micro instruction corresponding to the partial-width packed data instruction, the first micro instruction specifying a first operation and the second micro instruction specifying a second operation;

an execution unit to execute an operation specified by the first micro instruction on only a subset of the plurality of packed data elements; and

circuitry to eliminate the second micro instruction.

24. (Currently Amended) The processor of claim 23, wherein the decoder [[is a decoder]] comprises logic to create the second micro instruction by replicating the first micro instruction to create a replica and modifying the replica to create the second micro instruction.

25. (Currently Amended) The processor of claim 23, wherein the execution unit [[is an execution unit]] comprises logic to set a data element in a result packed data operand to a predetermined value.

Claims 26-32 (Cancelled)

33. (Currently Amended) A computer system comprising:

    a bus;

    a [[storage device including a]] flash memory coupled to the bus to store data;

    a processor coupled to the [[storage device]] flash memory by the bus to execute instructions;

    a memory of the processor to store a first packed data operand having a first plurality of data elements and a second packed data operand having a second plurality of data elements;

    a decoder of the processor coupled with the memory of the processor, the decoder to receive a partial-width packed data instruction and to decode the partial-width packed data instruction, wherein the [[partial width]] partial-width packed data instruction indicates the first packed data operand and the second packed data operand, and indicates a first operation to be performed on a subset of corresponding pairs of data elements of the first and the second packed data operands; and

[[a partial-width]] an execution unit of the processor coupled with the decoder to execute the operation on the subset of corresponding pairs of data elements.

34. (Currently Amended) The computer system of claim 33:

wherein the decoder is a decoder to convert the partial-width packed data instruction into a first micro instruction that corresponds to a first subset of at least one corresponding pair of data elements of the first and the second packed data operands and a second micro instruction that corresponds to a second subset of at least one corresponding pair of data elements of the first and the second packed data operands; and

wherein the [[partial-width]] execution unit is [[a partial-width execution unit]] to execute an operation specified by the first micro instruction on the first subset.

35. (Previously Presented) The computer system of claim 34:

wherein the processor is a processor to eliminate the second micro instruction; and

wherein the processor is a processor to set at least one result data element corresponding to the second subset to a predetermined value.

36. (Currently Amended) The computer system of claim 33:

further comprising a first port coupled with the memory to receive the first packed data operand and a second port coupled with the memory to [[substantially simultaneously]] receive the second packed data operand;

further comprising divide circuitry to divide the first packed data operand into a first subset comprising at least one data element and a second subset comprising

at least one data element and to divide the second packed data operand into a third subset comprising at least one data element and a fourth subset comprising at least one data element; and

wherein the [[partial-width]] execution unit is [[a partial-width execution unit]] to perform the first operation on at least one corresponding pair of data elements of the first and the third subsets to generate at least one resulting data element.

37. (Currently Amended) The computer system of claim 36:

further comprising delay circuitry to delay the second subset and to delay the fourth subset; and

wherein after the delay, the [[partial-width]] execution unit is [[a partial-width execution unit]] to perform the first operation on at least one corresponding pair of data elements of the second and the fourth subsets to generate at least one additional resulting data element.

38. (New) A method comprising:

receiving a partial-width packed data instruction, the partial-width packed data instruction specifying locations in a memory of a first packed data operand and a second packed data operand, the partial-width packed data instruction specifying generation of a packed data result, the packed data result having as one or more data elements one or more results of one or more operations performed on one or more pairs of data elements of the first and the second packed data operands, and the packed data result having as one or more remaining data elements one or more predetermined values; and

generating the packed data result.

39. (New) The method of claim 38, wherein the first and the second packed data operands comprise 128-bit operands, and wherein the packed data result comprises a 128-bit packed data result.
40. (New) The method of claim 39, further comprising dividing the operands into low and high order segments that each comprise 64-bits.
41. (New) The method of claim 38, further comprising storing the packed data result over the first packed data operand.
42. (New) The method of claim 38, wherein the data elements store floating point data.
43. (New) The method of claim 38, wherein said generating the packed data result comprises dividing each of the operands into separate segments and sequentially processing each segment using the same hardware.
44. (New) The method of claim 38, wherein said generating the packed data result comprises:  
generating a micro instruction; and  
using the micro instruction to access only lowest order portions of the first and the second packed data operands.
45. (New) The method of claim 38, wherein said generating the packed data result comprises:  
accessing full-widths of the first and the second packed data operands;  
dividing the operands into low-order and high-order segments; and

sequentially processing the low-order segments and the high-order segments.

46. (New) The method of claim 38, wherein the operation is one of add and multiply.

47. (New) The method of claim 38, wherein said generating the packed data result comprises:

determining a portion of the packed data result by performing an operation specified by the partial-width packed data instruction on only a subset of pairs of data elements of the first and the second packed data operands; and

setting another portion of the packed data result to the one or more predetermined values.

48. (New) The method of claim 38, wherein the one or more predetermined values comprise a value of a data element of the first packed data operand.

49. (New) The method of claim 48, wherein said generating the packed data result comprises passing value of a data element of the first packed data operand to the packed data result.

50. (New) The method of claim 38, wherein said generating the packed data result comprises clearing a data element of the packed data result.

51. (New) The method of claim 38, wherein the partial width packed data instruction comprises a scalar packed data instruction, and wherein the packed data result includes a result of an operation performed on only a single pair of data elements of the first and the second packed data operands

52. (New) The method of claim 38, further comprising converting the partial-width packed data instruction into a micro instruction capable of accessing only portions of the first and the second packed data operands.
53. (New) The method of claim 38, further comprising limiting reported exceptions to those connected with said determining the portion of the packed data result.
54. (New) The method of claim 38, further comprising reducing power consumption by selectively shutting down circuitry that is unnecessary to generate the packed data result.
55. (New) A method comprising:

receiving an instruction specifying generation of a 128-bit packed data result operand, the 128-bit packed data result operand having as a lowest order data element a result of an operation performed on lowest order data elements of a first 128-bit packed data operand and a second 128-bit packed data operand, and the 128-bit packed data result operand having as at least one remaining data element a value of a data element of the first 128-bit packed data operand; and

executing the instruction.
56. (New) The method of claim 55, wherein the instruction identifies two 128-bit logical source registers respectively having stored therein the first and the second 128-bit packed data operands.
57. (New) The method of claim 56, further comprising storing the 128-bit packed data result operand over the first 128-bit packed data operand in a first of the two 128-bit logical source registers.

58. (New) The method of claim 55, wherein the data elements store floating point data.
59. (New) The method of claim 55, wherein each of the data elements stores a single precision floating point value.
60. (New) The method of claim 55, wherein each of the data elements stores a double precision floating point value.
61. (New) The method of claim 55, wherein said executing the instruction includes dividing each of the operands into separate segments and sequentially processing each segment using the same hardware.
62. (New) The method of claim 55, wherein said executing the instruction comprises:  
generating a micro instruction; and  
using the micro instruction to access only lowest order portions of the first and the second 128-bit packed data operands.
63. (New) The method of claim 55, wherein said executing the instruction comprises:  
accessing full-widths of the first and the second 128-bit packed data operands;  
dividing the operands into low-order and high-order segments; and  
sequentially processing the low-order segments and the high-order segments.
64. (New) The method of claim 55, wherein the operation is one of add and multiply.

65. (New) A method comprising:

receiving a scalar packed data instruction, the scalar packed data instruction specifying locations in a 128-bit logical register file of a processor of a first 128-bit packed data operand and a second 128-bit packed data operand, each of the 128-bit packed data operands including a low-order segment and a high-order segment, and each of the segments including two 32-bit single precision floating point data elements, the instruction specifying generation of a 128-bit packed data result operand, the 128-bit packed data result operand having as a data element a result of an operation performed on a single pair of corresponding least significant data elements of the first and the second 128-bit packed data operands, and the 128-bit packed data operand having as one or more remaining data elements one or more predetermined values; and

generating the 128-bit packed data result operand according to the instruction.

66. (New) The method of claim 65, wherein the operation is selected from an add operation and a multiply operation.

67. (New) The method of claim 65, wherein the one or more predetermined values comprise one or more values selected from a data element of the first packed data operand and an identity function value.

68. (New) An apparatus comprising:

a register file to provide logical registers to store packed data operands, each of the packed data operands including multiple data elements;

a decoder to receive instructions including a partial-width packed data instruction that specifies the generation of a packed data result, the packed data result

including as a first data element a result of an operation performed on one pair of data elements of a first packed data operand and a second packed data operand, and the packed data result including as a second data element a predetermined value; and

an execution unit coupled with the decoder and the register file.

69. (New) The apparatus of claim 68, wherein the predetermined value comprises a value of a data element of the first packed data operand.
70. (New) The apparatus of claim 68, wherein the execution unit comprises logic to pass through a data element of one of the operands to the packed data result.
71. (New) The apparatus of claim 68, wherein the predetermined value comprises an identity function value.
72. (New) The apparatus of claim 68, wherein the execution unit comprises logic to zero a data element of the packed data result.
73. (New) The apparatus of claim 68, wherein the partial-width packed data instruction comprises a scalar packed data instruction.
74. (New) The apparatus of claim 68, wherein the first and the second packed data operands comprise 128-bit operands, and wherein the packed data result comprises a 128-bit operand.
75. (New) The apparatus of claim 68, wherein the partial-width packed data instruction specifies that the packed data result be stored over the first packed data operand in a logical source register of the register file.

76. (New) The apparatus of claim 68, wherein the data elements store floating point data.
77. (New) The apparatus of claim 68, wherein the execution unit comprises logic to sequentially process separate segments of the first and the second packed data operands.
78. (New) The apparatus of claim 68, wherein the execution unit comprises logic to generate the first data element and the second data element in a staggered manner.
79. (New) The apparatus of claim 78, wherein the execution unit comprises logic to use a micro instruction to access lowest order portions of the first and the second packed data operands.
80. (New) The apparatus of claim 78, wherein the execution unit comprises:  
logic to access full-widths of the operands from locations in a memory; and  
logic to process portions of the operands at different times.
81. (New) The apparatus of claim 68, wherein the operation is one of add and multiply.
82. (New) The apparatus of claim 68, further comprising logic to reduce power consumption by selectively shutting down circuitry of the execution unit that is unnecessary to generate the packed data result.
83. (New) The apparatus of claim 68, implemented in a computer system including a network coupling device.

84. (New) An apparatus comprising:

a register file to provide 128-bit logical registers to store packed data operands, each of the packed data operands including multiple data elements;

a decoder to receive instructions including a partial-width packed data instruction that specifies generation of a 128-bit packed data result operand, the 128-bit packed data result operand having as a lowest order data element a result of an operation performed on lowest order data elements of a first 128-bit packed data operand and a second 128-bit packed data operand, and the 128-bit packed data result operand having as one or more remaining data elements one or more values of one or more data elements of the first 128-bit packed data operand; and

an execution unit coupled with the decoder and the register file.

85. (New) The apparatus of claim 84, wherein the partial-width packed data instruction identifies two 128-bit logical source registers respectively having stored therein the first and the second 128-bit packed data operands.

86. (New) The apparatus of claim 85, wherein the partial-width packed data instruction specifies that the 128-bit packed data result operand be stored over the first 128-bit packed data operand in a first of the two 128-bit logical source registers.

87. (New) The apparatus of claim 84, wherein the data elements store floating point data.

88. (New) The apparatus of claim 87, wherein each of the data elements stores a single precision floating point value.

89. (New) The apparatus of claim 87, wherein each of the data elements stores a double precision floating point value.
90. (New) The apparatus of claim 84, wherein the execution unit comprises logic to sequentially process separate segments of the first and the second 128-bit packed data operands.
91. (New) The apparatus of claim 84, wherein the execution unit comprises logic to use a micro instruction to access only lowest order portions of the first and the second 128-bit packed data operands.
92. (New) The apparatus of claim 84, wherein the execution unit comprises logic to access full-widths of the first and the second 128-bit packed data operands, divide the operands into low-order and high-order segments, and sequentially process the low-order segments and the high-order segments.
93. (New) The apparatus of claim 84, wherein the operation is one of add and multiply.
94. (New) The apparatus of claim 84, wherein each of the 128-bit logical registers is provided for using a 128-bit physical register.
95. (New) The apparatus of claim 84, wherein each of the 128-bit logical registers is provided for using two 64-bit physical registers.
96. (New) An apparatus comprising:
  - a 128-bit logical register file of a processor to store 128-bit packed data operands;
  - a decoder to receive and decode instructions including a scalar packed data instruction, the scalar packed data instruction specifying locations in the 128-bit

logical register file of a first 128-bit packed data operand and a second 128-bit packed data operand, each of the 128-bit packed data operands including a low-order segment and a high-order segment, and each of the segments including two 32-bit single precision floating point data elements, the scalar packed data instruction specifying generation of a 128-bit packed data result operand, the 128-bit packed data result operand having as a data element a result of an operation performed on a single pair of corresponding least significant data elements of the first and the second 128-bit packed data operands, and the 128-bit packed data result operand having as one or more remaining data elements one or more predetermined values; and

an execution unit coupled with the decoder, the execution unit to execute the instructions including the scalar packed data instruction.

97. (New) The apparatus of claim 96, wherein the operation is selected from an add operation and a multiply operation.

98. (New) The apparatus of claim 96, implemented in a computer system including a flash memory.

99. (New) An apparatus comprising:

    a bus;

    a flash memory coupled with the bus; and

    a processor coupled with the bus, the processor including:

        a decoder to receive a partial-width packed data instruction; and

an execution unit coupled with the decoder, the execution unit to generate a packed data result according to the partial-width packed data instruction, the packed data result including as a first data element a result of an operation specified by the instruction performed on a pair of data elements of a first and a second packed data operands, and the packed data result including as a second data element a predetermined value.

100. (New) The apparatus of claim 99, wherein the predetermined value comprises a value of a data element of an operand.
101. (New) The apparatus of claim 99, wherein the predetermined value comprises an identity function value.
102. (New) The apparatus of claim 99, wherein the execution unit comprises logic to generate the first data element and the second data element in a staggered manner.
103. (New) The apparatus of claim 102, wherein the execution unit comprises: logic to access full-widths of the operands from locations in a memory; and logic to process portions of the operands at different times.
104. (New) The apparatus of claim 102, wherein the decoder comprises logic to convert the partial-width packed data instruction into a micro instruction that if executed accesses and processes only portions of the packed data operands.
105. (New) The apparatus of claim 99, wherein the execution unit comprises logic to pass through a data element of one of the operands to the packed data result.

106. (New) The apparatus of claim 99, wherein the execution unit comprises logic to zero a data element of the packed data result.
107. (New) The apparatus of claim 99, wherein the partial-width packed data instruction comprises a scalar packed data instruction.
108. (New) The apparatus of claim 99, wherein the first and the second packed data operands comprise 128-bit packed data operands.
109. (New) The apparatus of claim 99, further comprising logic to reduce power consumption by selectively shutting down circuitry of the execution unit that is unnecessary to generate the packed data result.
110. (New) A method comprising:
  - receiving an instruction specifying locations in a memory of a first packed data operand and a second packed data operand;
  - accessing full-widths of the first packed data operand and the second packed data operand from the memory;
  - dividing the first packed data operand into a first portion and a second portion;
  - dividing the second packed data operand into a third portion and a fourth portion;
  - determining a first partial result by performing an operation specified by the instruction on the first and the third portions; and
  - determining a second partial result by processing the second and the fourth portions, wherein determining the second partial result includes setting at least one data element to a predetermined value,

wherein the same hardware is used to determine the first and the second partial results.

111. (New) The method of claim 110, wherein the instruction comprises a scalar packed data instruction.

112. (New) The method of claim 110, further comprising setting a portion of the first partial result to a value selected from a value of a data element of an operand and an identify function value.

113. (New) The method of claim 110, wherein dividing the operands into the portions comprises dividing the operands into low and high order segments.

114. (New) The method of claim 113:  
wherein the operands comprise 128-bit operands; and  
wherein the low and high order segments each comprise 64-bit segments.

115. (New) The method of claim 110, further comprising, prior to determining the second partial result, delaying the second and the fourth portions.

116. (New) The method of claim 110, further comprising reducing power consumption by shutting down circuitry that is not necessary to determine the second partial result.

117. (New) The method of claim 110, further comprising:

delaying the first partial result;  
after said delaying the first partial result, collecting the first and the second partial results; and

writing the partial results to a destination specified by the instruction.

118. (New) An apparatus comprising:

divide logic to receive and divide a full-width of a first packed data operand specified by a partial-width packed data instruction into a first portion and a second portion, and to receive and divide a full-width of a second packed data operand specified by the partial-width packed data instruction into a third portion and a fourth portion;

delay elements coupled with the divide logic to receive and delay the first portion and the third portion; and

an execution unit coupled with the divide logic and the delay elements, the execution unit to provide a packed data result specified by the partial-width packed data instruction to a destination specified by the partial-width packed data instruction, the packed data result including one or more results of operations performed on only a subset of the data elements of the first and the second packed data operands.

119. (New) The apparatus of claim 118, wherein the first and the second packed data operands comprise 128-bit operands, and wherein the divide logic comprises logic to divide the 128-bit operands into 64-bit low order segments and 64-bit high order segments.

120. (New) The apparatus of claim 118, wherein the same circuitry of the execution unit processes the second and the fourth portions at a different time than the first and the third portions.

121. (New) The apparatus of claim 118, further comprising a delay element to delay a portion of the packed data result corresponding to the second and the fourth portions.
122. (New) The apparatus of claim 118, wherein the execution unit comprises logic to determine a first portion of the packed data result by performing an operation selected from an add operation and a multiply operation on a subset of data elements of the second and the fourth portions and to set a second portion of the packed data result to one or more predetermined values.
123. (New) The apparatus of claim 118, wherein the instruction comprises a scalar packed data instruction, and wherein the execution unit comprises logic to perform an operation specified by the instruction on a single pair of corresponding data elements of the second and the fourth portions.
124. (New) The apparatus of claim 118, further comprising logic to reduce power consumption by shutting down circuitry of the execution unit that is unnecessary to process the first and the third portions.
125. (New) The apparatus of claim 118, implemented in a computer system including a flash memory.
126. (New) The apparatus of claim 118, implemented in a computer system including a network coupling device.
127. (New) A method comprising:  
receiving an instruction specifying a first packed data operand and a second packed data operand, the instruction specifying an operation to be performed on

only a subset of corresponding pairs of data elements of the first and the second packed data operands;

converting the instruction into one or more micro instructions including a first micro instruction;

receiving only a portion of the first and the second packed data operands using the first micro instruction;

generating a result by processing the portion of the first and the second packed data operands; and

providing the result to a destination specified by the instruction.

128. (New) The method of claim 127, wherein converting the instruction into the one or more micro instructions further comprises converting the instruction into a second micro instruction.
129. (New) The method of claim 128, further comprising eliminating the second micro instruction.
130. (New) The method of claim 128, wherein converting comprises replicating an operation and then modifying the replicated operation in order to generate the second micro instruction.
131. (New) The method of claim 127, wherein the instruction comprises a scalar packed data instruction.
132. (New) The method of claim 127, wherein the first packed data operand and the second packed data operand each comprise 128-bit packed floating point data operands.

133. (New) An apparatus comprising:

a decoder to receive an instruction specifying a first packed data operand and a second packed data operand, the instruction specifying an operation to be performed on only a subset of corresponding pairs of data elements of the first and the second packed data operands;

logic of the decoder to convert the instruction into one or more micro instructions, the one or more micro instructions including a first micro instruction; and

an execution unit coupled with the decoder, the execution unit to receive a portion of the first and the second packed data operands that is specified by the first micro instruction, the execution unit to provide a result associated with the first micro instruction to a destination specified by the instruction.

134. (New) The apparatus of claim 133, wherein the logic to convert the instruction into the one or more micro instructions comprises logic to convert the instruction into the first micro instruction and a second micro instruction, and wherein the execution unit includes logic to process the first and the second micro instructions at different times using the same hardware.

135. (New) The apparatus of claim 134, wherein the execution unit comprises logic to eliminate the second micro instruction.

136. (New) The apparatus of claim 133, wherein the instruction comprises a scalar packed data instruction.

137. (New) The apparatus of claim 133, wherein the operands comprise 128-bit packed floating point data operands.

138. (New) The apparatus of claim 133, implemented in a computer system including a flash memory.

139. (New) The apparatus of claim 133, implemented in a computer system including a network coupling device.

140. (New) An apparatus comprising:

a bus;

a processor coupled to the bus;

a flash memory coupled to the bus;

partial-width packed data instructions stored in the flash memory; and

an execution unit of the processor, the execution unit to provide a packed data result according to a partial-width packed data instruction,

the packed data result including as a first data element a result of an operation specified by the instruction performed on a pair of data elements of a first and a second packed data operands, and

the packed data result including as a second data element a predetermined value.

141. (New) The apparatus of claim 140, wherein the partial-width packed data instructions each specify one or more logical registers including 128-bit operands.

142. (New) The apparatus of claim 140, wherein the partial-width packed data instructions each specify an operation to be performed on only a subset of data elements of the operands.

143. (New) The apparatus of claim 140, wherein the partial-width packed data instruction comprises a scalar packed data instruction.

144. (New) The apparatus of claim 140, further comprising logic to selectively shut down circuitry of the processor not needed to execute the partial-width packed data instruction.